Docket No. 49986-0506 *Patent*

UNITED STATES PATENT APPLICATION

FOR

AUTOMATED MANAGEMENT OF DEVELOPMENT PROJECT FILES OVER A NETWORK

INVENTOR:

TETSURO MOTOYAMA

PREPARED BY:

HICKMAN PALERMO TRUONG & BECKER, LLP
1600 WILLOW STREET
SAN JOSE, CALIFORNIA 95125
(408) 414-1080

EXPRESS MAIL CERTIFICATE OF MAILING

"Express Mail" mailing label number EL734779394US

Date of Deposit JUNE 13, 2001

AUTOMATED MANAGEMENT OF DEVELOPMENT PROJECT FILES OVER A NETWORK

FIELD OF THE INVENTION

[0001] The present invention relates generally to project management and, more specifically, to automating tasks for management of a development project over a network.

BACKGROUND OF THE INVENTION

[0002] Computer software development projects are inherently difficult to manage. This difficulty is partly due to the large number of tasks and associated deliverables that comprise a software package and the vastness of paperwork and project files associated with these tasks and deliverables. Another contributing factor are the complex interdependencies established between individual tasks and deliverables during the development cycle of a software package.

[0003] Management of development projects typically includes organizing, maintaining, and controlling access to project documents, schedules, and the like. Furthermore, there are often multiple development projects occurring concurrently within an enterprise organization, thus significantly expanding the document management efforts. Historically, management of a master project schedule entails, among other tasks, manually entering data into a scheduling application, manually creating links between schedules, and manually aggregating individual developers' task schedules into the master project schedule. These are cumbersome and error-prone tasks, with little to no oversight and quality control.

[0004] A master project schedule is often in a state of flux, whereby management solicits the developers for task statuses and related schedule updates. Often, the feedback provided to management by the developers has little oversight and is not

according to a rigid policy, procedure, or verification process. Thus, the actual status of a project schedule is often difficult to ascertain since the progress of individual tasks are dictated by subjective, and often self-supporting, progress reports by those individuals that are assigned to the task.

[0005] For example, some scheduling systems allow a developer to signify that a task is partially completed, i.e., ninety percent completed. This information is then entered into the scheduling system to determine whether the project is on-schedule. However, because there is generally no accountability as to whether an individual's status is reliable, the current process of obtaining project status tends to shadow the realistic progress of the project.

[0006] In view of the foregoing, there is a clear need for a technique for automating management of a development project that reduces manual tasks related to document management and schedule tracking, and which includes criteria for schedule tracking that ensures quality thereof.

SUMMARY OF THE INVENTION

[0007] A technique is provided for managing a project schedule for a development project based on automatic aggregation of individual task schedules, where the individual task schedules are automatically updated based on inspection results from two or more inspectors specified to inspect a project task product. The schedules, and consequently the updates thereof, are governed by a policy specifying that a task cannot be partially completed. The inspection results are linked to the individual task schedules, which are linked to the associated project schedule, whereby they are made available to authorized persons over a network.

[0008] Another technique is provided for managing project files over a network. Using this technique, a project is proposed through an on-line project initiation form. Acceptance of the project proposal triggers the creation of individual sites for each individual specified to contribute to the project, where individual task schedules and draft project files can be linked to the individual site. Furthermore, the individual sites are linked to a project site and associated file directories are automatically created and also linked to the project site. Upon passing inspection by two or more specified inspectors, a draft file is changed to an official file and the official file is denoted accordingly.

[0009] These automated techniques provide advantages over prior manual processes, including but not limited to, ensuring the quality and integrity of and controlling access to project files and schedules, and minimizing work-load and potential errors related to management and manipulation of the project files and schedules.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] The present invention is illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings and in which like reference numerals refer to similar elements and in which:

[0011] FIG. 1 illustrates an example of an operating environment in which aspects of the invention can be implemented;

[0012] FIG. 2 is a block diagram that illustrates a computer system upon which embodiments of the invention can be implemented;

[0013] FIG. 3 illustrates examples of data components of a database and web components, according to an embodiment of the invention;

[0014] FIG. 4A is a flow chart illustrating steps for initiating automated management of project files over a network, according to an embodiment of the invention;

[0015] FIG. 4B is a flowchart illustrating steps an individual (client) performs in relation to the document inspection process, according to an embodiment of the invention;

[0016] FIG. 4C is a flowchart illustrating a server-side process continuing from R1 of FIG. 4B, according to an embodiment of the invention;

[0017] FIG. 4D is a flowchart illustrating the server-side process continuing from R2 of FIG. 4B; according to an embodiment of the invention;

[0018] FIG. 5A is a block diagram of an arrangement for initiating a project via an on-line interactive form, according to an embodiment of the invention;

[0019] FIG. 5B illustrates an example of a printed or displayed project initiation form that can be utilized in an embodiment of the invention;

[0020] FIG. 5C is a continuation of the exemplary printed/displayed project initiation form of FIG. 5B;

[0021] FIG. 6 illustrates an example of a project site, according to an embodiment of the invention;

[0022] FIG. 7A illustrates an example of a project document index, according to an embodiment of the invention;

[0023] FIG. 7B illustrates link relationships between an example web index page, the project database, and electronic or physical files/objects managed by the database, according to an embodiment of the invention;

[0024] FIG. 8 is a flow chart illustrating steps for managing project files over a network, according to an aspect of the invention;

[0025] FIG. 9 illustrates an embodiment of the method of FIG. 8, wherein individual task schedules are automatically managed;

[0026] FIG. 10 illustrates another embodiment of the method of FIG. 8, wherein a summary management schedule is automatically managed;

[0027] FIG. 11 is a block diagram illustrating associations utilized to manage a project schedule, according to an aspect of the invention;

[0028] FIG. 12 illustrates an example of an individual task schedule, according to an embodiment of the invention;

[0029] FIG. 13 illustrates an printed or displayed example of an on-line inspection form which is utilized to automatically update an individual task schedule, according to an embodiment of the invention;

[0030] FIG. 14 illustrates an example of a management schedule, according to an embodiment of the invention; and

[0031] FIG. 15 is a flow chart illustrating steps for generating and updating a schedule for a project, according to an aspect of the invention.

DETAILED DESCRIPTION OF THE INVENTION

[0032] Techniques for automating tasks involved in the management of a development project are described. The techniques are described herein primarily in reference to a software development project, but those skilled in the art should recognize that the benefits of the invention are also available when applying the techniques to other development projects. In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent, however, that the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to avoid unnecessarily obscuring the present invention.

[0033] OPERATING ENVIRONMENT

[0034] FIG. 1 illustrates an example of an operating environment in which aspects of the invention can be implemented. The exemplary operating environment comprises a plurality of workstations 102, a web server 104, and a database 106, all connected directly or indirectly to a software development network 108 for communication therebetween. Optionally, a database 110 may be present for reasons described below.

[0035] Workstations 102 are typically computer systems configured as illustrated by the computer system 200 of FIG. 2, and are utilized, for example, by the software engineers/developers to complete tasks associated with a development project. Pertinent non-limiting examples of such tasks include initiating projects, preparing and maintaining task schedules, designing software architecture, creating specifications, creating software code, implementing and testing software code, inspecting various task products, etc. In addition, project managers utilize workstations 102 for accessing information to review and manage the progress of

the project. The developers and managers transmit communications through the network 108 to the other connected components, i.e., web server 104 and database 106.

[0036] Web server 104 depicts a conventional web server, which is a program that, using the appropriate protocols (e.g., Hypertext Transfer Protocol [HTTP] and Transmission Control Protocol/Internet Protocol [TCP/IP]), serves the files that form web pages (e.g., Hypertext Markup Language [HTML] or Extensible Markup Language [XML] files), to users, such as developers or managers at a workstation 102. In general, the majority of information exchanged and managed during the development project life cycle is served by the web server 104 over the network 108. Furthermore, aspects of the techniques for automating management of development project files, as described herein, may be implemented and executed on the web server 104, although practice of the invention is not limited to such an implementation. The techniques could also be implemented on any other processing system, such as workstation 102 or a similarly configured computer system as illustrated in FIG. 2.

[0037] Database 106 depicts a conventional database for storing information related to the development project, thus providing access to the information by authorized individuals at workstations 102 or web server 104, through queries transmitted over the network 108. The type of information stored on database 106 is virtually limitless, examples including project initiation forms, individual and aggregated management task schedules, specifications, software code, inspection reports, web page files, and document directories and indexes. In addition, other information may be stored on the database 106, as illustrated in and described in reference to FIG. 3. In alternative operating environments, a conventional database 110 is connected directly to the network 108 as a database server.

[0038] Network 108 depicts a conventional network, e.g., a packet-switched network, for facilitating the exchange of information between and among various connected components, such as workstation 102, web server 104, and database 106. The network 108 may be a Local Area Network (LAN), such as a conventional Ethernet, Fast Ethernet, a token ring, or a wireless LAN such as specified in 802.11a and 802.11b (developed by a working group of the Institute of Electrical and Electronics Engineers [IEEE]), which may be implemented within an enterprise. In addition, network 108 may also be a Wide Area Network (WAN), such as the Internet, for facilitating communication with remote users through a Virtual Private Network (VPN), or the network 108 may represent a combination of a LAN and a WAN. In addition, network 108 can be formed using a variety of different mediums, including but not limited electrical wire or cable, optical, or wireless connections.

[0039] HARDWARE OVERVIEW

[0040] FIG. 2 is a block diagram that illustrates a computer system 200 upon which embodiments of the invention can be implemented. Computer system 200 additionally illustrates an example of a system configuration of the workstation 102 (FIG. 1) and the web server 104 (FIG. 1). Computer system 200 includes a bus 202 or other communication mechanism for communicating information, and a processor 204 coupled with bus 202 for processing information. Computer system 200 also includes a main memory 206, such as a random access memory (RAM) or other dynamic storage device, coupled to bus 202 for storing information and instructions to be executed by processor 204. Main memory 206 also may be used for storing temporary variables or other intermediate information during execution of instructions to be executed by processor 204. Computer system 200 further includes a read only memory (ROM) 208 or other static storage device coupled to bus 202 for

storing static information and instructions for processor 204. A storage device 210, such as a magnetic disk, optical disk, or magneto-optical disk, is provided and coupled to bus 202 for storing information and instructions.

[0041] Computer system 200 may be coupled via bus 202 to a display 212, such as a cathode ray tube (CRT) or a liquid crystal display (LCD), for displaying information to a computer user. An input device 214, including alphanumeric and other keys, is coupled to bus 202 for communicating information and command selections to processor 204. Another type of user input device is cursor control 216, such as a mouse, a trackball, or cursor direction keys for communicating direction information and command selections to processor 204 and for controlling cursor movement on display 212. This input device typically has two degrees of freedom in two axes, a first axis (e.g., x) and a second axis (e.g., y), that allows the device to specify positions in a plane.

[0042] The invention is related to the use of computer system 200 for implementing the techniques described herein. According to one embodiment of the invention, those techniques are performed by computer system 200 in response to processor 204 executing one or more sequences of one or more instructions contained in main memory 206. Such instructions may be read into main memory 206 from another computer-readable medium, such as storage device 210. Execution of the sequences of instructions contained in main memory 206 causes processor 204 to perform the process steps described herein. In alternative embodiments, hard-wired circuitry may be used in place of or in combination with software instructions to implement the invention. Thus, embodiments of the invention are not limited to any specific combination of hardware circuitry and software.

[0043] The term "computer-readable medium" as used herein refers to any medium that participates in providing instructions to processor 204 for execution.

Such a medium may take many forms, including but not limited to, non-volatile media, volatile media, and transmission media. Non-volatile media includes, for example, optical, magnetic disks, or magneto-optical disks, such as storage device 210. Volatile media includes dynamic memory, such as main memory 206. Transmission media includes coaxial cables, copper wire and fiber optics, including the wires that comprise bus 202. Transmission media can also take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0044] Common forms of computer-readable media include, for example, a floppy disk, a flexible disk, hard disk, magnetic tape, or any other magnetic medium, a CD-ROM, any other optical medium, punchcards, papertape, any other physical medium with patterns of holes, a RAM, a PROM, and EPROM, a FLASH-EPROM, any other memory chip or cartridge, a carrier wave as described hereinafter, or any other medium from which a computer can read.

[0045] Various forms of computer readable media may be involved in carrying one or more sequences of one or more instructions to processor 204 for execution. For example, the instructions may initially be carried on a magnetic disk of a remote computer. The remote computer can load the instructions into its dynamic memory and send the instructions over a telephone line using a modem. A modem local to computer system 200 can receive the data on the telephone line and use an infra-red transmitter to convert the data to an infra-red signal. An infra-red detector can receive the data carried in the infra-red signal and appropriate circuitry can place the data on bus 202. Bus 202 carries the data to main memory 206, from which processor 204 retrieves and executes the instructions. The instructions received by main memory 206 may optionally be stored on storage device 210 either before or after execution by processor 204.

[0046] Computer system 200 also includes a communication interface 218 coupled to bus 202. Communication interface 218 provides a two-way data communication coupling to a network link 220 that is connected to a local network 222. For example, communication interface 218 may be an integrated services digital network (ISDN) card or a modem to provide a data communication connection to a corresponding type of telephone line. As another example, communication interface 218 may be a local area network (LAN) card to provide a data communication connection to a compatible LAN. Wireless links may also be implemented. In any such implementation, communication interface 218 sends and receives electrical, electromagnetic or optical signals that carry digital data streams representing various types of information.

[0047] Network link 220 typically provides data communication through one or more networks to other data devices. For example, network link 220 may provide a connection through local network 222 to a host computer 224 or to data equipment operated by an Internet Service Provider (ISP) 226. ISP 226 in turn provides data communication services through the world wide packet data communication network now commonly referred to as the "Internet" 228. Local network 222 and Internet 228 both use electrical, electromagnetic or optical signals that carry digital data streams. The signals through the various networks and the signals on network link 220 and through communication interface 218, which carry the digital data to and from computer system 200, are exemplary forms of carrier waves transporting the information.

[0048] Computer system 200 can send messages and receive data, including program code, through the network(s), network link 220 and communication interface 218. In the Internet example, a server 230 might transmit a requested code

for an application program through Internet 228, ISP 226, local network 222 and communication interface 218.

[0049] The received code may be executed by processor 204 as it is received, and/or stored in storage device 210, or other non-volatile storage for later execution. In this manner, computer system 200 may obtain application code in the form of a carrier wave.

[0050] PROJECT DATABASE

[0051] FIG. 3 illustrates example of data components of the database 106 and web components. Database 106 can store files representing various project documents, some being generic project-related documents employed by an enterprise for providing guidance with respect to administering and controlling development projects, and some being specific to a particular project. For examples of generic documents, the database can be configured to store one or more template forms 302 for use by project participants (i.e., engineer/developers, managers, and others), such as a project initiation form (see FIGs. 5B and 5C for a printed example of an interactive project initiation form) or an inspection form (see FIG. 13 for a printed example of an interactive inspection form); and one or more manuals 304, policies 306, and procedures 308, for instructing project participants on enterprise infrastructure, policy and procedures, at least with respect to development projects. The forms 302 facilitate the interactive input of information into the system database 106 and are primarily used by the clients, or individual project participants, and also define the printed outputs.

[0052] Project data 310 refers to project-specific documents that may include, but is not limited to, completed project initiation forms (see FIGs. 5B and 5C), individual task schedules (see FIG. 12), aggregated management task schedules (see FIG. 14), specifications, software code, completed inspection forms (see FIG. 13), web page

files, and document directories and indexes. Note that the document directories and indexes may alternatively or additionally be stored on the database 110 (FIG. 1). A project participant working on a workstation 102, or alternatively on web server 104, can utilize a search engine 320 to access the database 106 and search for the various generic and project-specific documents.

[0053] Different levels of project-specific information can be accessed from the database 106, as is depicted by a projects home page 312 and one or more project sites 314. The projects home page 312 provides links to the one or more project sites 314. As is known in the art, a link is a selectable connection from one word, picture, or information object to another. One example of an implementation of a link is a hyperlink, utilizing a suitable protocol and language such as HTTP and HTML, respectively. The links allow a user to access the project sites 314 from the home page 312, by enacting the link. The link is enacted typically through use of the cursor control 216 (FIG. 2) and/or the input device 214 (FIG. 2), by interacting with an appropriate application such as a conventional web browser. Examples of the information that is linked to, and thus accessible from, the project sites 314 are described below.

[0054] INITIATING AUTOMATED PROJECT FILE MANAGEMENT

[0055] FIG. 4A is a flow chart illustrating steps for initiating automated management of project files over a network, according to an embodiment of the invention. First, at step 402, an individual completes a project initiation form for submission to management for project approval. FIG. 5A is a block diagram of an arrangement for initiating a project via an on-line interactive form 500. The interactive form can be an HTML or XML based page. The project initiator enters the necessary information, for example, the project title 502, project description 504, anticipated project members and responsibilities 506, overall schedule 508, and

budget 510, through a network interface, such as a web browser. The entered information is transmitted through the web server 104 to the database 106, where it is initially stored in draft form prior to approval of the requested project, at step 404 of FIG. 4A. Furthermore, the initiator can revise the draft form until proceeding to the project authorization/approval process. The blocks of information depicted in database 106 in FIG. 5A comprise the various information that can be extracted and presented in a project home page or site, such as site 600 of FIG. 6.

[0056] An example of a printed or displayed project initiation form 550, illustrating pertinent information headings with the information excluded, is illustrated in FIGs. 5B and 5C. The interactive project initiation form 500 (as exemplified also as form 550) is available from the form 302 (FIG. 3) component of database 106 (FIG. 1), and is linked to other data in the database 106 for automated entry of some data fields. The format of the printed/displayed project initiation form 550 is associated with the form 302.

[0057] Referring back to FIG. 4A, at decision block 406 it is determined whether the appropriate project approval authority has approved the proposed project. If the project is not approved, then at step 408 the project initiation form is marked to indicate that the project is not approved, and is stored as a draft in the database 106. Upon approval of the proposed project, at step 410 the project is automatically assigned a project number and an official project site, such as project site 314 (FIG. 3), is automatically created and linked to the project home page, such as project home page 312 (FIG. 3). In addition, various database 106 entries and index pages are created at step 410. At step 412, web sites for individual project participants are created; and linked to the appropriate official project site, at step 414. In addition, the necessary entries are created in the database 106, and linked to the appropriate skeleton files that are utilized by the project participants. The project participants

are able to link any working or draft documents to their individual site, whereby the documents will be available to authorized project members through the appropriate chain of links from the project home page, i.e., projects home page 312. Access permissions are restricted and controlled according to document control policies. In certain embodiments, a directory of project files associated with the project is created, stored in the database 106 and database 110 (FIG. 1), and linked to the project site (as shown in FIG. 6). Sub-directories and indexes can be created, as applicable, and linked to the appropriate directory entry on the project site.

[0058] INSPECTION PROCESS - CLIENT

[0059] FIG. 4B is a flowchart illustrating steps an individual (client) performs in relation to the document inspection process, according to one embodiment. At step 452, a document, record, or report (reference material) is generated by an individual project participant. At the decision block of step 454, it is determined whether the reference material is the type that requires inspection to ensure quality. For example, some information, such as technical information and records, may not require the quality inspection. In this case, at step 456, the individual requests registration of the document, whereby the flow goes to R1, which is illustrated in FIG. 4C. If inspection of the reference material is deemed necessary to ensure quality, the individual arranges inspection thereof, at step 458. At step 460, an inspection is requested whereby it is registered in the project file management system, to be managed by the processes described herein. In this case, the flow goes to R2, which is illustrated in FIG. 4D.

[0060] INSPECTION PROCESS – SERVER

[0061] FIG. 4C is a flowchart illustrating the server-side process continuing from R1 of FIG. 4B, where the reference material is not inspected and where document registration is requested. At step 472, the reference material is copied into database

106 (FIG. 1) under a controlled environment so that the individual who created the material is no longer able to modify it. At step 474, information entries, for example, the title, date, and originator, are created in the appropriate index page (see 700 of FIG. 7A). At step 476, links are created from the index page to the appropriate documents and corresponding document fields.

[0062] FIG. 4D is a flowchart illustrating the server-side process continuing from R2 of FIG. 4B, where an inspection of the reference material is requested. Generally, the file management system embodying the file management processes described herein, monitors whether the requested inspection completes on schedule and thus, whether inspection results are available. At step 482, an inspection result object (see 774 of FIG. 7B) is linked to the reference material. At decision block 484, it is periodically determined whether inspection results are available. If inspection results are not yet available, the process essentially waits for a certain period to pass, at step 485, and then returns to step 484 to look for inspection results. Once inspection results are available, they are analyzed. At decision block 486, the inspection result is analyzed to determine whether the reference material is denoted as accepted by the associated inspector. If it is determined that the material is not accepted, then at step 488 it is determined whether the material requires re-inspection or if it is conditionally accepted (a disposition different than accepted). If the material requires re-inspection, the database is updated accordingly. At this point, the process could return to step 484 to determine if additional inspection results are available. If the material is conditionally accepted, the flow goes to decision block 490, where the material is checked to determine whether it is modified to meet the specified conditions from the inspection report, and whether the corrections are certified by an inspection chief. If the material is not yet certified

by the inspection chief, at step 491, the process waits for a period and returns to step 490.

[0063] Once it is determined at decision block 490 that the material has been certified by the inspection chief, the process continues to step 492, which is the same step that is performed if the reference material is accepted at step 486. At step 492, the current inspection material is copied into database 106 (FIG. 1) under a controlled environment so that the individual who created the material is no longer able to modify it. At step 494, information entries, for example, the title, date, and originator, are created in the appropriate index page (see 700 of FIG. 7A). At step 496, links are created from the index page to the appropriate documents and corresponding document fields.

[0064] PROJECT WEB PAGES

[0065] FIG. 6 illustrates an example of a project site 600, with links (underlined entities) to several other pages of information specific to a particular project (in this case, the J06 project as shown at top of 600). Links include, but are not limited to, a directory 602 of official project documents and records; a link to project source code 604; a link to the project requirements 606; a link to a project schedule 608; a link to one or more current task lists 610 and member web sites 612 of individuals, i.e., engineers/developers, working on the project.

[0066] The directory 602 presents links to indexes of various official documents and records associated with the project, for non-limiting examples, project documents, inspection results, meeting records, changes, error tracking, and other records.

[0067] The project schedule link 608 provides access to the aggregated management task schedule, which is exemplified in and described in reference to FIG. 14. The current task list link 610 provides access to the task schedules of each

individual assigned tasks for the project, which is exemplified in and described in reference to FIG. 12. Furthermore, the relation between the individual task schedules, accessed via link 610, and the aggregated management task schedule, accessed via link 608, is described in detail below under the headings "Management Schedule Generation" and "Updating a Project Schedule." Finally, the member website link 612 provides access to one or more individual web sites, the creation of which is described above in reference to step 408 of FIG. 4. The individual web sites provide access to draft documents being worked on by the individual, as well as to the individual task lists (see FIG. 12).

[0068] FIG. 7A illustrates an example of an index 700, which is an "Index of Project Documents." The index 700, and any other indexes linked to the directory 602, includes links to actual files stored in the database 106 (FIG. 1). FIG. 7B illustrates the link relationships between an example web index page 750, the project database 106, and electronic or physical files/objects 770 managed by the database 106. A reference number 752, a document title 754, and a revision number 756 are linked to the same object, that is, official documents 772. Effective date 758 corresponds to the date that the document is effective or last revised in correspondence with the revision number 756. An inspection 760 field is linked to the inspection results 774 corresponding to the inspections performed on the document. Status 762 presents the current status of the indexed document, that is whether the document is still awaiting authorization from the appropriate party. The status 762 field is linked to the authorization history object 776. The information included in the index 750, under the management of the database 106, is displayable to a client through a web browser interface.

[0069] METHOD FOR MANAGING PROJECT FILES OVER A NETWORK

[0070] FIG. 8 is a flow chart illustrating steps for managing project files over a network, according to an aspect of the invention. Project initiation information, preferably including at least a description of the project and of individuals performing a project task, is received at step 802. An example of the information received at step 802 is illustrated in the project initiation form 550 of FIGs. 5B and 5C. At step 804, it is determined whether the project has been approved by the appropriate entity. At step 806, if the project is approved, the project initiation information is stored in a database such as database 106 (FIG. 1 and FIG. 3) in a manner that indicates that the project is approved.

[0071] At step 808 of FIG. 8, an individual web site, or page, is created for each of the individuals working on the project. Various informational and work product documents can be linked to the individual web sites; for example, draft files and task schedules. Access to this information is typically regulated and is available to authorized individuals over a network. At step 810, the individual sites are linked to the project site, as exemplified by member web site 612 of project site 600 (FIG. 6). A directory of files associated with the project is created at step 812, which is stored in the database at step 814. The directory of files is linked to the project site at step 816, as exemplified by directory 602 and described in the associated text in reference to FIG. 6.

[0072] At step 818 of FIG. 8, it is determined whether a draft file is completed, where completion is defined as passing inspection from at least two inspectors. This criteria which defines a completed file serves at least the following purposes: it ensures the quality and integrity of the file, as a result of the inspection by two or more persons related to the project but not the creators of the file; and it clarifies whether a task is completed by providing a binary completion status, i.e., the task is

recorded as completed or not without allowing recordation of a percentage of completion.

[0073] Upon completion of a draft file, the status of the file is changed from draft to official and it is stored in the database indicating its status, at step 820. Finally, at step 822, the official file is linked to the project site. As exemplified in reference to FIG. 6 and FIG. 7, the official file may be indirectly linked to the project site through a link to an index, such as index 700, which is in turn linked to a directory, such as directory 602, which is presented on the project site, such as site 600.

[0074] FIG. 9 illustrates an embodiment of the method of FIG. 8, wherein individual task schedules are automatically managed. At step 902, one or more task schedules are received from individuals working on the project. At step 904, the individual task schedules are stored in the database 106 (FIG. 1). Steps 906 and 908, which can complete in any order, include at step 906, automatically linking the individual task schedule to the associated individual's site which is accessible via link 610 of FIG. 6, and at step 908, automatically linking the individual task schedule to the project site. The project site link is depicted as current task list 610 and described in the associated text in reference to FIG. 6.

[0075] MANAGEMENT SCHEDULE GENERATION

[0076] FIG. 10 illustrates another embodiment of the method of FIG. 8, wherein a summary management schedule is automatically managed. At step 1002, one or more task schedules are received from individuals working on the project. At step 1004, a management schedule associated with the same project is updated based on the individual task schedules. An advantage of having the individual task schedules linked to the management task schedule is that the management task schedule can automatically update upon changes to the individual task schedules.

At step 1006, the management task schedule is linked to the project site, as depicted as project schedule 608 and described in the associated text in reference to FIG. 6.

[0077] FIG. 11 is a block diagram illustrating link associations that are used to manage a project schedule by automatically updating the project schedule, according to an aspect of the invention. In this example, multiple task lists 1102, 1104, and 1106 (see FIG. 12 for an example of an individual task list/schedule) are linked to a system manager 1110, which embodies methods for managing the project schedule as described herein. The task lists 1102-1106 are for individuals working on the project, and each task list is typically associated with a separate individual, although practice of the invention is not so limited. The project schedule 1112 (or "management schedule") is an aggregation of the individual task lists 1102-1106, and typically does not include task definitions at the level of detail as the individual task schedules. Hence, the management schedule 1112 summarizes the individual task schedules 1102, 1104, and 1106. For example, FIG. 14 illustrates an example of a management schedule 1400, which is described in further detail below.

[0078] The completion of each task in the individual task schedules is linked to inspection forms, completed versions of which are stored in the database 106 (FIG. 1). For example, FIG. 13 illustrates an example of a printed or displayed inspection form 1300, which is described in further detail below. Upon a positive disposition by two or more individuals that have inspection authority with respect to an individual task product, the associated task is considered completed. In one embodiment, the status of each task is binary variable, in that a task cannot be recorded as partially completed. For example, according to the project file management techniques described herein, a task is either completed or not completed, but not a percentage completed. Thus, only after the authorized task inspectors have completed their inspections of the task product and have each

stored a completed inspection form in the database 106, does the task product receive an "accepted," or similar, disposition. In certain embodiments, the individual task schedule is automatically updated based on the results of the completed inspection forms. In addition, once the individual task schedules 1102, 1104, and 1106 are updated, the management schedule 1112 is subsequently updated based on the updated individual task schedules 1102, 1104, and 1106.

[0079] FIG. 12 illustrates an example of an individual task schedule 1200 for an individual "TM." FIG. 13 illustrates an example of a printed or displayed on-line inspection form 1300. In one embodiment, individual task schedule 1200 and inspection form 1300 provide status data that is used to automatically generate or update a management schedule 1400, as illustrated in FIG. 14. This process is facilitated by the links described in reference to FIG. 11. Upon completion of an inspection form (such as form 1300), an individual task schedule (such as task schedule 1200) is updated according to the completed inspection form, and a management schedule (such as schedule 1400) is consequently updated.

[0080] Referring to FIG. 13, document reference 1302 is mapped to the same task in the related individual task schedule. Note that the document reference 1302 does not refer to documents only, but more generally to products of individual tasks. Furthermore, a result reference 1304 is mapped to the "Actual End" column 1208 (FIG. 12) of the related individual task schedule, such as schedule 1200 of FIG. 12. The date that is automatically entered into the "Actual End" column 1208 is automatically determined based on the latest acceptance completion date for all of the required inspection forms (i.e., "Accept" in result reference 1304) for a particular task. The method includes logic for determining when all of the inspections are completed, and whether all of the completed inspection forms have indicated

"Accept" in result reference 1304, in order to determine the "Actual End" date for column 1208.

[0081] Referring to FIG. 12 and FIG. 14, certain cells of the task schedule 1200 are mapped to the management schedule 1400. For example, cell 1202 of task schedule 1200, which is associated with the earliest "Planned Start" of a particular task, is mapped to cell 1402 of management schedule 1400. Similarly, cell 1204, which is associated with the latest "Planned End" of a particular task, is mapped to cell 1404. Thus, if data in cell 1202 or 1204 is added, revised, or deleted, cell 1402 or 1404 is automatically revised accordingly. Cells 1206 and 1406 are similarly related to each other as are the cells previously discussed. There are numerous other task schedule cells similar to cell 1202 that map to associated management schedule cells for a particular individual, in this case, "TM," thus providing automatic updates of the high-level management schedule 1400 according to an aspect of the invention.

[0082] UPDATING A PROJECT SCHEDULE

[0083] FIG. 15 is a flow chart illustrating steps for generating and/or updating a schedule for a project, according to an aspect of the invention. At step 1502, a completed inspection form including the inspection data is received over a network from the database 106 (FIG. 1). A completed inspection form corresponds to each of two or more inspectors specified to inspect a task product, whereby the completed inspection form includes information based on the inspection. Refer back to FIG. 13 for an example of an inspection form 1300.

[0084] At step 1504, an individual's task schedule (i.e., the individual responsible for completing the task), is automatically updated based on the received inspection forms. According to a policy, a project task is not completed unless all inspection result reports so indicate. At step 1506, a management schedule, as exemplified in FIG. 14 and which is an aggregation of all of the individual task schedules associated

with the project, is automatically updated based on the updated individual task schedules from step 1504.

[0085] In one embodiment, the individual and management schedules are governed by a policy specifying that a project task cannot be partially completed and the automatic updating of the schedules is performed according to this policy.

[0086] At step 1508, the completed project task product is stored in the database 106 (FIG. 1 and FIG. 3) and access to the product is regulated according to a document control policy. In one embodiment, the completed task products are accessible via a packet-based network, such as the Internet or an enterprise network, via appropriate links such as hyperlinks.

[0087] Hence, the foregoing detailed description describes techniques for automated management of development project files over a network. In addition, in this disclosure, certain process steps are set forth in a particular order, and alphabetic and alphanumeric labels are used to identify certain steps. Unless specifically stated in the disclosure, embodiments of the invention are not limited to any particular order of carrying out such steps. In particular, the labels are used merely for convenient identification of steps, and are not intended to imply, specify or require a particular order of carrying out such steps.

[0088] In the foregoing specification, the invention has been described with reference to specific embodiments thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader spirit and scope of the invention. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.
